

# **MIFARE® PLUS DEMO MANUAL**

*Written in both C# and C++ (WxWidgets)*

Link: <https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-mfp-examples-csharp-gui.git>

C++: <https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-mfp-examples-cpp-gui.git>

## **1 - About Mifare® Plus cards**

- Keys can be stored as MIFARE® CRYPTO1 keys (6 bytes per sector) and as AES keys (16 bytes per sector)
- Memory structure identical to MIFARE® Classic 4K (sectors, blocks)

## **2 - Application demo**

When you simply double click on the .exe and the program starts, it automatically searches for the device that is connected via USB cable. When the device is found, it will print out "Communication port is opened" in lower left corner and reader description in the text box on the bottom of the application. After successful port open, it will start continuous search for card in its NFC field. If found - it will print out cards type and UID in upper left corner. If not, it will

show UFR\_NO\_CARD in the same place where cards type would be. For example:

Mifare Plus example

Card information

Card type: DL\_MIFARE\_PLUS\_S\_4K\_SL1 Card UID: 04:45:91:0A:78:23:80:00:00:00

Auth mode

☒ Provided AES key ☐ Reader AES key

Personalization Auth SL1 Switch to SL3 Change Master Key Change Config Key Change Sector key Field Config Set Get UID Change VC polling t

Card master key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF SL1 card authentication key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Card configuration key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF Select VC key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Level 2 switch key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF Proximity check key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Level 3 switch key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF VC polling ENC key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

VC polling MAC key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF Personalize card

All keys must be 16 bytes long.

Info about uFReader:

Serial number: UN101055

Communication port opened

From the above picture, we can see all of the tabs dedicated to working with MIFARE<sup>®</sup> Plus cards. Keep in mind that not all functions mentioned in this demo work with all three cards security levels. Also, all functions on these tabs, except for the last three - “Data read, data write, Write keys into reader” depend on Authentication mode that can be chosen in upper right corner. If we check “Provided AES key” then we will use functions with provided key required. And vice versa, if we check “Reader AES key” then our functions will use reader key that we select on our tabs.

Mifare Plus example

Card information

Card type: DL\_MIFARE\_PLUS\_S\_4K\_SL1 Card UID: 04:45:91:0A:78:23:80:00:00:00

Auth mode

☒ Provided AES key ☐ Reader AES key

Personalization Auth SL1 Switch to SL3 Change Master Key Change Config Key Change Sector key Field Config Set Get UID Change VC polling t

Let's start from Personalization tab:

Personalization	Auth SL1	Switch to SL3	Change Master Key	Change Config Key	Change Sector key	Field Config Set	Get UID	Change VC polling	⏪ ⏩
Card master key:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF			SL1 card authentication key:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
Card configuration key:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF			Select VC key:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
Level 2 switch key:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF			Proximity check key:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
Level 3 switch key:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF			VC polling ENC key:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
VC polling MAC key:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF			Personalize card					

All keys must be 16 bytes long.

On this tab we have 9 keys that the card uses for its intended use and purpose. Inputting all of them and clicking the “Personalize card” button will have our card set up for future use. All of these keys have their use explained in their names. For example “Card configuration key” is used for changing cards configuration settings, such as Field configuration settings, authorization of changing VC polling ENC/MAC keys, reader AES keys, etc..

Cards default security level is Security Level 0. Cards security level automatically updates to level 1 as soon as we complete card Personalization. On this level we can also choose AES authentication, which is by default used on SL3. We accomplish that on the next tab.

Personalization	Auth SL1	Switch to SL3	Change Master Key	Change Config Key	Change Sector key
Provided AES key					
Enter SL1 card authentication key: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF					
Reader AES key					
Enter reader key index for SL1 card authentication (0 - 15): 0 ▾					
AES authentication on SL1					

On this tab, depending on what authentication mode we selected in the upper right corner, we use one of these keys to enable MIFARE® Plus authentication with AES keys instead of CRYPTO1 while it's in SL1 mode. MIFARE® Plus supports all MIFARE® Classic value-block operations either in SL1 or SL3 mode.

After security upgrade to SL3 mode, MIFARE® Plus uses Advanced Encryption Standard (AES) for authentication, data integrity and encryption. In our demo we switch to SL3 on the next tab - "Switch to SL3".

Personalization Auth SL1 Switch to SL3 Change Master Key Change Config Key

Provided AES key

Enter level 3 switch key:

FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Reader AES key

Enter reader key index for level 3 switch key (0 - 15)

0

Switch to SL3

Usage of this tab is same for the previous. We check the authentication mode in the upper right corner, enter or select key that we will use to authenticate this process and click on the button.

On next tab, we can change cards Master key. To do so, we must enter previous, old master key so we can authorize usage of the new key that we enter.

Personalization
Auth SL1
Switch to SL3
Change Master Key
Change Config Key

Provided AES key

Old card master key
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

New card master key
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Reader AES key

Reader key index for old card master key: (0 - 15)
0

New card master key
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Change master key

The same principle goes for all of the key changes mentioned in our demo with an exception of changing sector key - where we have to specify for which sector we shall be changing AES key.

As for the Field configuration settings, in other words use choice of Random UID that Desfire cards have or regular UID, we authorize that setting with our cards Configuration key and select one of our options on top: Use random UID or standard UID:

Personalization
Auth SL1
Switch to SL3
Change Master Key
Change Config Key
Change Sector key
Field Config Set
Get UID

Random ID option

☐ use random ID
☒ use UID

Provided AES key

Configuration key:
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Reader AES key

Enter reader configuration key index (0 - 15):
0

Field configuration set

The random UID feature means that the cards UID is new every time it is powered on, which means every time it is used. To randomise the UID means that everytime the card is used, a new UID is recorded. This each activity of the card cannot be linked to the user, and he

chain of activity cannot be traced to the card. Card needs to be in SL3 mode to use this feature. Only way to get cards true UID with this feature enabled is to get UID with usage of VC polling ENC & MAC keys.

We will demonstrate this in next tab "Get UID".

For example, our card generates a random UID like this:

#1

 Mifare Plus example

Card information	
Card type:	DL_MIFARE_PLUS_S_4K_SL3
Card UID:	08:9C:C7:37:00:00:00:00:00

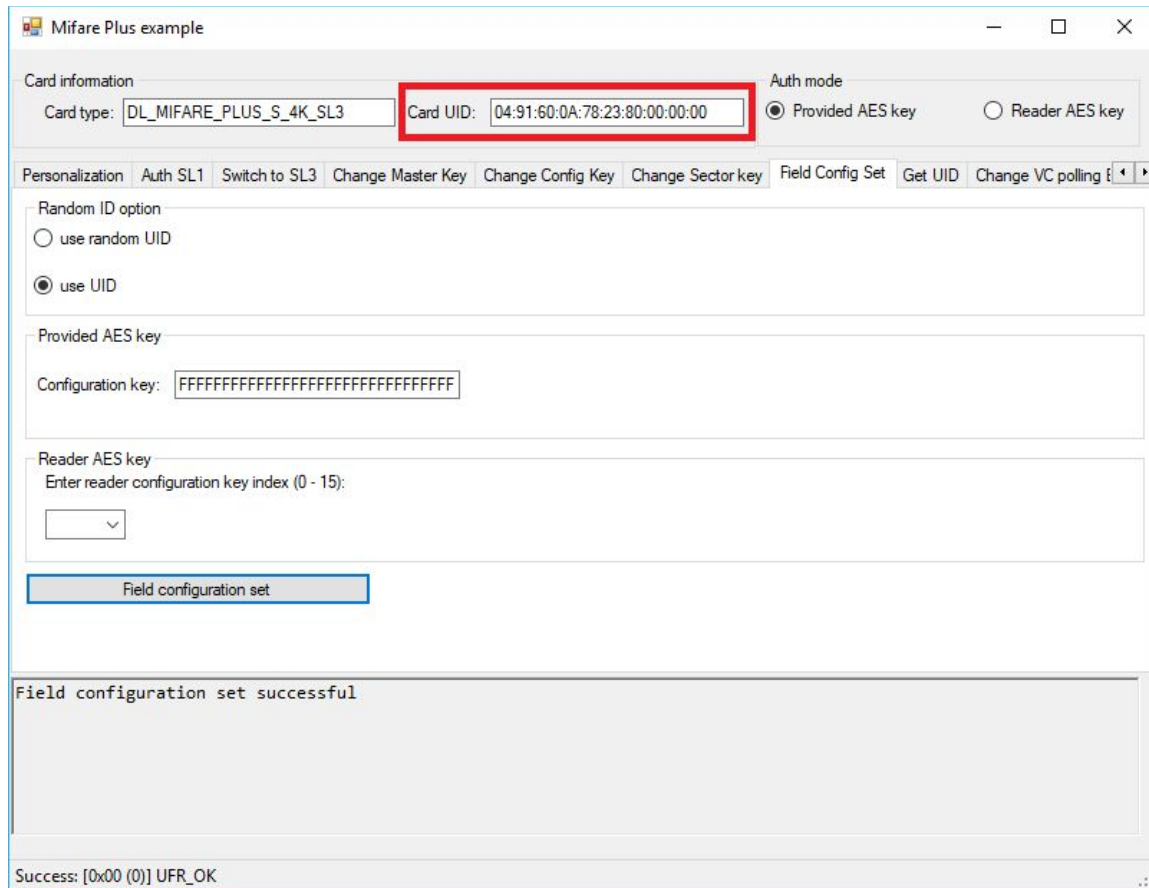
#2

 Mifare Plus example

Card information	
Card type:	DL_MIFARE_PLUS_S_4K_SL3
Card UID:	08:9C:C7:37:00:00:00:00:00

Each time our card gets into NFC field and gets powered on - it shall get new randomized UID.

To see cards true/factory UID, we enter ENC and MAC key or select them by their index if they are stored in the reader.



From the image above we can see that UID has changed and will be so until we enable random UID again. From the text box below we can see that our settings were set successfully.

Now we shall demonstrate Data reading and writing. This section has an exception of different authentication modes and different reading/writing modes influenced by our choice of previously mentioned authentication modes.

Here we have Key A and Key B authentication modes, as for the modes of reading/writing we have Provided key mode, Reader key mode, AKM1 mode and AKM2 mode. For reference of these settings such as KeyA/B authentications and AKM1/2 mode, you can find explanations in our UFR Series NFC reader API here:

<https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-doc>



Example for Block read:

The screenshot shows the 'Mifare Plus example' application window. At the top, 'Card information' includes 'Card type: DL\_MIFARE\_PLUS\_S\_4K\_SL1' and 'Card UID: 04:14:59:0A:78:23:80:00:00:00'. The 'Auth mode' is set to 'Provided AES key'. A menu bar contains options like 'Change Config Key', 'Change Sector key', 'Field Config Set', 'Get UID', 'Change VC polling ENC key', 'Change VC polling MAC key', 'Data read', 'Data write', and 'V'. Below the menu, the 'Block read' tab is active. It contains four sub-sections: 'Provided key', 'Reader key', 'AKM1 key', and 'AKM2 key'. Each section has 'Authentication mode' (Key A or Key B), a 'Block' dropdown (set to 0), and a 'Key' field. The 'Provided key' field contains 'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'. The 'Reader key' section has a 'Reader key index' dropdown (set to 0) and a note: '(0 - 31 for CRYPTO 1) and (0 - 15 for AES) key'. The 'Block read reader key' button is highlighted. Below these sections, a status bar shows 'Block read was successful.' and 'Block data: 04:14:59:0A:78:23:80:18:42:00:04:01:11:00:39:11'. At the bottom, a success message reads 'Success: [0x00 (0)] UFR\_OK'.

Here we see that we successfully read block 0 with one of these functions, and block data has been written in the bottom of the screen.

Similar principle goes for Data writing with exception of entering data we wish to write in our card, for example Block write:



Mifare Plus example

Card information

Card type: DL\_MIFARE\_PLUS\_S\_4K\_SL1 Card UID: 04:45:91:0A:78:23:80:00:00

Auth mode

☒ Provided AES key ☐ Reader AES key

Change Config Key Change Sector key Field Config Set Get UID Change VC polling ENC key Change VC polling MAC key Data read Data write

Block write Block in sector write Linear write

Provided key

Authentication mode

☒ Key A ☐ Key B Block: 1

Key: FFFFFFFFFFFFFFFFFFFFFFFFFF

Block write provided key

Reader key

Authentication mode

☒ Key A ☐ Key B Block: 1

Key index: 0 (0 - 31 for CRYPTO 1) and (0 - 15 for AES) key

Block write reader key

AKM1 key

Authentication mode

☒ Key A ☐ Key B Block: 1

Block write AKM1 key

AKM2 key

Authentication mode

☒ Key A ☐ Key B Block: 1

Block write AKM2 key

Block: (0 - 128 for 2K cards) and (0 - 255 for 4k cards)

Data: 000102030405060708090A0B0C0D0E0F

Block write was successful.

Success: [0x00 (0)] UFR\_OK

We just choose block that we wish to write into, enter or select key if needed and click on the provided button.

If we go back to reading, we will see our data written in,for example, block 1 successfully.

Mifare Plus example

Card information

Card type:  Card UID:

Auth mode

☒ Provided AES key ☐ Reader AES key

Change Config Key

Change Sector key

Field Config Set

Get UID

Change VC polling ENC key

Change VC polling MAC key

Data read

Data write

W

Block read

Block in sector read

Linear read

Provided key

Authentication mode

☒ Key A ☐ Key B

Block:

Key:

Block read provided key

Reader key

Authentication mode

☒ Key A ☐ Key B

Block:

Reader key index:  (0 - 31 for CRYPTO 1) and (0 - 15 for AES) key

Block read reader key

AKM1 key

Authentication mode

☒ Key A ☐ Key B

Block:

Block read AKM1 key

AKM2 key

Authentication mode

☒ Key A ☐ Key B

Block:

Block read AKM2 key

Block: (0 - 128 for 2K cards) and (0 - 255 for 4k cards)

Block read was successful.

Block data: 00:01:02:03:04:05:06:07:08:09:0A:0B:0C:0D:0E:0F

Success: [0x00 (0)] UFR\_OK

For storing keys in our reader, we have dedicated tab “Write keys into reader” where we can enter either CRYPTO1 or AES keys, and lock further key input in our device or unlock it.

Mifare Plus example

Card information

Card type:  Card UID:

Auth mode

☒ Provided AES key ☐ Reader AES key

Change Sector key | Field Config Set | Get UID | Change VC polling ENC key | Change VC polling MAC key | Data read | Data write | **Write keys into reader**

CRYPTO 1 Keys

CRYPTO 1 key:

Enter key index (0 - 31):

AES keys

AES key:

Enter key index (0 - 15):

Unlock reader

Password:

Lock reader

Password:

CRYPTO 1 key writing was successful

Success: [0x00 (0)] UFR\_OK

As a reminder: CRYPTO1 key is 6 bytes long, AES key is 16 bytes long, and our password input is 8 characters, for the sake of simplicity.

Git repos of this example:

<C# link>

<WxWidgets link>

API reference for functions used in this demo:

<https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-doc>